

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:	§	Examiner: Fennema, Robert E.
Mitchell Alsup, et al.	§	
	§	
	§	Group Art Unit: 2183
Serial No. 10/726,902	§	
	§	Atty. Dkt. No.: 5500-88700
Filed: December 3, 2003	§	
	§	
For: Transitioning from	§	
Instruction Cache to Trace	§	
Cache on Label Boundaries	§	

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

In response to the Notice of Panel Decision mailed December 6, 2007, Appellants present this Appeal Brief. **No extension of time fee is due since this Appeal Brief is filed within one month of the mailing date of the Notice of Panel Decision.** Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

I. REAL PARTY IN INTEREST

As evidenced by the assignment recorded at Reel/Frame 014767/0579, the subject application is owned by Advanced Micro Devices, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at One AMD Place, P.O. Box 3453, Sunnyvale, CA 94088.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-35 are pending in the application and stand finally rejected. The rejection of claims 1-35 is being appealed. A copy of claims 1-35 is included in the Claims Appendix herein below.

IV. STATUS OF AMENDMENTS

No amendments have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a microprocessor that includes an instruction cache, a branch prediction unit, a trace cache, and a prefetch unit coupled to the instruction cache, the branch prediction unit, and the trace cache. (*See, e.g.*, FIG. 1, microprocessor 100, instruction cache 106, branch prediction unit 132, trace cache 160, and prefetch unit 108; and p. 7 lines 3-5 and 18-27.)

The instruction cache is configured to store instructions. (*See, e.g.*, p. 7, lines 18-18-22; and p. 8, lines 15-16.)

The trace cache is configured to store a plurality of traces of instructions. (*See, e.g.*, FIG. 2, trace cache entry 162; p. 10, lines 18-29; and p. 13, lines 6-9.)

The prefetch unit is configured to fetch instructions from the instruction cache until the branch prediction unit outputs a predicted target address. (*See, e.g.*, p. 15, lines 5-15; FIG. 3, block 301, and the feedback loop from the negative exit of decision block 303 to block 301; and p. 15, line 27 – p. 16, line 3.)

The prefetch unit is configured to check the trace cache for a match for the predicted target address in response to the branch prediction unit outputting the predicted target address. (*See, e.g.*, FIG. 3, the positive exit of decision block 303, and block 305; and p. 16, lines 5-9.)

The prefetch unit is configured to not check the trace cache for a match until the branch prediction unit outputs the predicted target address. (*See, e.g.*, FIG. 3, the negative exit of decision block 303 to block 301; p. 15, line 27 – p. 16, line 3; and p. 18, lines 9-23.)

In response to the prefetch unit identifying a match for the predicted target address in the trace cache, the prefetch unit is configured to fetch one or more of the

traces from the trace cache. (*See, e.g.*, FIG. 3, the positive exit of decision block 305, and block 307; and p. 16, lines 9-15.)

Independent claim 14 is directed to a computer system that includes a system memory and a microprocessor coupled to the system memory. (*See, e.g.*, FIG. 1, microprocessor 100 coupled to system memory 200; and p. 7, lines 18-28.)

The microprocessor includes an instruction cache, a branch prediction unit, a trace cache, and a prefetch unit coupled to the instruction cache, the branch prediction unit, and the trace cache. (*See, e.g.*, FIG. 1, microprocessor 100, instruction cache 106, branch prediction unit 132, trace cache 160, and prefetch unit 108; and p. 7 lines 3-5 and 18-27.)

The instruction cache is configured to store instructions. (*See, e.g.*, p. 7, lines 18-18-22; and p. 8, lines 15-16.)

The trace cache is configured to store a plurality of traces of instructions. (*See, e.g.*, FIG. 2, trace cache entry 162; p. 10, lines 18-29; and p. 13, lines 6-9.)

The prefetch unit is configured to fetch instructions from the instruction cache until the branch prediction unit outputs a predicted target address. (*See, e.g.*, p. 15, lines 5-15; FIG. 3, block 301, and the feedback loop from the negative exit of decision block 303 to block 301; and p. 15, line 27 – p. 16, line 3.)

The prefetch unit is configured to check the trace cache for a match for the predicted target address in response to the branch prediction unit outputting the predicted target address. (*See, e.g.*, FIG. 3, the positive exit of decision block 303, and block 305; and p. 16, lines 5-9.)

The prefetch unit is configured to not check the trace cache for a match until the branch prediction unit outputs the predicted target address. (*See, e.g.*, FIG. 3, the

negative exit of decision block 303 to block 301; p. 15, line 27 – p. 16, line 3; and p. 18, lines 9-23.)

In response to the prefetch unit identifying a match for the predicted target address in the trace cache, the prefetch unit is configured to fetch one or more of the traces from the trace cache. (*See, e.g.*, FIG. 3, the positive exit of decision block 305, and block 307; and p. 16, lines 9-15.)

Independent claim 27 is directed to a method that includes receiving a retired instruction. (*See, e.g.*, p. 12, line 27- p. 13, line 3; p. 16, lines 26-29; and p. 17, lines 11-13.)

The method also includes determining if a previous trace under construction duplicates a trace in a trace cache and if the received instruction corresponds to a branch label. (*See, e.g.*, FIG. 4, decision blocks 353 and 357; and p. 19, lines 18-27.)

The method further includes, in response to determining that a previous trace under construction duplicates a trace in a trace cache and that the received instruction corresponds to a branch label, beginning construction of a new trace. (*See, e.g.*, FIG. 4, the positive exits from decision blocks 353 and 357, and block 359; and p. 19, lines 29-31.)

Independent claim 32 is directed to method that includes fetching instructions from an instruction cache. (*See, e.g.*, p. 15, lines 5-15; FIG. 3, block 301 and p. 15, line 27-28.)

The method includes continuing to fetch instructions from the instruction cache without searching a trace cache until a branch target address is generated. (*See, e.g.*, p. 15, lines 5-15; FIG. 3, block 301, and the feedback loop from the negative exit of

decision block 303 to block 301; FIG. 3, the negative exit of decision block 303 to block 301; p. 15, line 27 – p. 16, line 3; and p. 18, lines 9-23.)

The method further includes, in response to a branch target address being generated, searching a trace cache for an entry corresponding to the branch target address. (*See, e.g.*, FIG. 3, the positive exit of decision block 303, and block 305; and p. 16, lines 5-9.)

Independent claim 35 is directed to a microprocessor that includes means for receiving a retired operation. (*See, e.g.*, FIG. 1, trace generator 170, which is configured to receive basic blocks of retired operations from retire queue 102; p. 12, line 27 - p. 13, line 3; p. 16, lines 26-29; and p. 17, lines 11-13.)

The microprocessor also includes means for determining if a previous trace under construction duplicates a trace in a trace cache and if the received operation is a first operation at a branch label. (*See, e.g.*, FIG. 1 trace generator 170; FIG. 4, decision blocks 353 and 357; and p. 19, lines 4-16 and 18-27.)

The microprocessor further includes means for starting a new trace in response to determining that a previous trace under construction duplicates a trace in a trace cache and that the received operation is a first operation at a branch label. (*See, e.g.*, FIG. 1, trace generator 170; FIG. 4, the positive exits from decision blocks 353 and 357, and block 359; and p. 19, lines 29-31.)

The summary above describes various examples and embodiments of the claimed subject matter; however, the claims are not necessarily limited to any of these examples and embodiments. The claims should be interpreted based on the wording of the respective claims.

VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 2, 11-15 and 24-26 stand finally rejected under 35 U.S.C. § 102(b) as being anticipated by Rotenberg, et al. (hereinafter “Rotenberg”).

2. Claims 3 and 16 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Rotenberg in view of Patterson, et al. (hereinafter “Patterson”).

3. Claims 4, 10, 17 and 23 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Rotenberg in view of Braught, and further in view of Xia.

4. Claims 5-8 and 18-21 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Rotenberg, Xia, Braught and further in view of Lange, et al. (U.S. Patent 3,896,419) (hereinafter “Lange”).

5. Claims 9 and 22 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Rotenberg, Xia and Braught in view of Akkary, et al. (U.S. Patent 6,247,121) (hereinafter “Akkary”).

6. Claims 27-31 and 35 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Rotenberg, Xia, Braught and Lange in view of Akkary.

7. Claims 32 - 34 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Rotenberg in view of Xia.

VII. ARGUMENT

First ground of rejection:

The Examiner rejected claims 1, 2, 11-15 and 24-26 under 35 U.S.C. § 102(b) as being anticipated by Rotenberg, et al. (hereinafter “Rotenberg”). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 2, 12, 14, 15, and 25:

1. The cited art clearly fails to disclose *wherein the prefetch unit is configured to check the trace cache for a match for the predicted target address in response to the branch prediction unit outputting a predicted target address; and wherein the prefetch unit is configured to not check the trace cache for a match until the branch prediction unit outputs the predicted target address.*

In the system of Rotenberg, “The trace cache is accessed in parallel with the instruction cache and BTB using the current fetch address...” (emphasis added.) **It is clear in this and other passages previously cited by Appellants that in Rotenberg every fetch address is compared to the entries in the trace cache, regardless of the operation of the branch prediction unit.** Rotenberg teaches searching for a hit in the trace cache on every cycle, and then fetching from the instruction cache if there is not a hit in the trace cache.

By contrast, Appellants’ claim 1 recites *wherein the prefetch unit is configured to not check the trace cache for a match **until** the branch prediction unit outputs the predicted target address.* **In other words, the trace cache is not checked for a match on every cycle, but only on cycles for which the branch prediction unit outputs a predicted target address.** The Examiner previously submitted, “It is inherent that you

cannot check for a value if you do not know what the value is. The trace cache as disclosed by Rotenberg cannot search for the predicted target address in the cache if it does not know what the predicted target address is, and in fact, no cache or any other type of hardware can find something when it doesn't know what it is looking for, thus, the value must have been output from the branch prediction unit prior to the checking." Appellants assert that the Examiner's interpretation of the operation of Rotenberg is **demonstrably incorrect** since Rotenberg explicitly checks for a match on every cycle, regardless of the operation of the branch prediction unit and whether it outputs a predicted target address.

In the Final Action mailed July 18, 2007, the Examiner submitted, "When looking at the entire claim in context, it can be seen that the claim is referring to searching the cache for a match for the predicted target address, and that it cannot search the cache for the predicted target address until the predicted target address is generated" (emphasis Examiner's). **The Examiner has misread the claim.** The referenced limitation of the claim does not recite "not check the trace cache for the match" (i.e., referring to the match for the predicted target address of the previous limitation), but instead recites "not check the trace cache for a match" (i.e., any match between the current address and the trace cache). **The plain language of the claim does not support the Examiner's interpretation.** As explicitly recited in claim 1, multiple instructions are fetched from the instruction cache, **without checking for a match (hit) in the trace cache, until the branch prediction unit outputs a predicted branch address** (i.e., until a branch instruction is encountered for which the branch prediction unit outputs a predicted target address). Only then (**in response to this output**) is the trace cache checked for a match.

Appellants again assert that one of ordinary skill in the art having benefit of Appellants' disclosure could not agree with the Examiner's interpretation of the referenced limitation. The Examiner's repeated assertions that it is impossible to search for something that has not yet been generated are irrelevant to Appellants' argument and to what is actually recited in the claim. The Examiner's interpretation is also completely inconsistent with the specification (see, e.g., FIG. 3 and pages 15-18). Also, in remarks

in the Final Action regarding claim 32, **the Examiner admits that Rotenberg does not teach not searching the trace cache until a branch target address is generated,** instead teaching that the trace cache is searched on every instruction.

For at least the reasons above, Rotenberg clearly does not anticipate the above-referenced limitations of Appellants' claim.

Claim 14 includes limitations similar to claim 1, and so the arguments presented above apply with equal force to this claim, as well.

Claims 11 and 24:

1. The cited art clearly fails to disclose *wherein each of the plurality of traces comprises partially-decoded instructions*, as recited in claim 11.

The Examiner submits that it is inherent that a trace comprises a partially-decoded instruction, otherwise the necessary control information as shown in section 2.2 of Rotenberg would not be available. Appellants assert, however, that the control information described in section 2.2 is control information added to a trace cache entry when the trace cache entry is generated and stored along with the instructions themselves in the trace cache. For example, Rotenberg describes that the control information is “similar to the tag array of standard caches but contains additional state information.” The control information described in section 2.2 comprises control information generated by the branch predictor and other logic of the trace cache, not information decoded from the instruction itself. **Therefore, the Examiner's assertion that traces inherently comprise partially-decoded instructions is clearly incorrect.** Nothing in Rotenberg describes that the instructions in a trace entry must be partially decoded in order to generate the control information, nor would generating the control information described in section 2.2 require such decoding.

For at least the reasons above, the rejection of claim 11 is unsupported by the cited art, and removal thereof is respectfully requested.

Claim 24 includes limitations similar to claim 11, and so the arguments presented above apply with equal force to this claim, as well.

Claims 13 and 26:

1. The cited art clearly fails to disclose *wherein each of the plurality of traces is associated with a flow control field comprising a label for an instruction to which control will pass for each branch operation comprised in that trace, as recited in claim 13.*

The Examiner submits that Rotenberg teaches this limitation in Section 2.2, and that the “trace target address” and “trace fall-through address” are labels that describe where control will flow based on each branch operation, based on a prediction. However, these addresses are actually described as corresponding to the next fetch addresses if the last branch in the trace is predicted taken or not taken, respectively. There is nothing in Rotenberg that describes labels for an instruction to which control will pass for each branch operation comprised in the trace, as recited in claim 13, only for the last branch in the trace.

For at least the reasons above, the rejection of claim 13 is unsupported by the cited art, and removal thereof is respectfully requested.

Claim 26 includes limitations similar to claim 13, and so the arguments presented above apply with equal force to this claim, as well.

Second ground of rejection:

The Examiner rejected claims 3 and 16 under 35 U.S.C. § 103(a) as being unpatentable over Rotenberg in view of Patterson, et al. (hereinafter “Patterson”). Appellants traverse this rejection for at least the following reasons.

1. The cited art clearly fails to teach or suggest *wherein the branch prediction unit is configured to output the predicted target address in response to detection of a branch misprediction, as recited in claim 3.*

The Examiner admits that Rotenberg fails to disclose this limitation and relies on Patterson to teach it. The Examiner submits that Patterson teaches, “in order to reduce the penalty for a misprediction, you can fetch both the taken and not taken instructions, and put them in the BTB, which would then be able to immediately output the correct path on a misprediction without having to fetch it (Pages 276-277).” This passage actually describes reducing the penalty for a misprediction by fetching instructions from both the predicted and unpredicted directions, e.g., by using a dual-ported cache, an interleaved cache, or by fetching from one path and then the other (i.e., fetching from both paths before determining whether a branch has been mispredicted). It does not describe outputting a predicted target address (to be checked against entries in a trace cache, as in Appellants’ claims) in response to detection of a branch misprediction.

Claim 16 includes limitations similar to claim 3, and so the arguments presented above apply with equal force to this claim, as well.

Third ground of rejection:

The Examiner rejected claims 4, 10, 17 and 23 under 35 U.S.C. § 103(a) as being unpatentable over Rotenberg in view of Braught, and further in view of Xia. Appellants

traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 4 and 17:

For at least the reasons presented herein regarding the claims from which they depend, the rejection of claims 4 and 17 is unsupported by the cited art, and removal thereof is respectfully requested.

Claims 10 and 23:

1. The cited art clearly fails to teach or suggest *wherein the trace generator is configured to generate traces in response to instructions being decoded, as recited in claim 10.*

The Examiner submits that Rotenberg teaches this limitation in Section 2.2, “the trace can not be completed (written into the cache) until the trace target addresses are calculated, which in turn requires the instructions to be decoded.” Appellants assert that this does not teach the above-referenced limitation of Appellants’ claim, in which traces are generated (not just completed) in response to instructions being decoded. There is nothing in Rotenberg that describes trace generation being performed in response to instructions being decoded.

2. The Examiner has not provided any motivation or other reason to combine the references in teaching the specific limitations of claims 10. Therefore, the Examiner has failed to establish a *prima facie* obviousness of the claimed invention.

For at least the reasons above, the rejection of claim 10 is unsupported by the cited art, and removal thereof is respectfully requested.

Claim 23 includes limitations similar to claim 10, and so the arguments presented above apply with equal force to this claim, as well.

Fourth ground of rejection:

The Examiner rejected claims 5-8 and 18-21 under 35 U.S.C. § 103(a) as being unpatentable over Rotenberg, Xia, Braught and further in view of Lange, et al. (U.S. Patent 3,896,419) (hereinafter “Lange”). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 5 and 18:

1. The cited art clearly fails to teach or suggest *wherein the trace generator is configured to check the trace cache for a duplicate copy of the trace that the trace generator is constructing*, as recited in claim 5.

The Examiner admits that Rotenberg, Xia, and Braught fail to teach this limitation and relies on Lange to teach it. As discussed below regarding claims 27 and 35, Lange’s description of the operation of a data cache during fetching from main memory has absolutely nothing to do with the specific limitations of claim 5. Checking Rotenberg’s trace cache for a duplicate trace before collecting a new trace is also contradictory to the Examiner’s remarks regarding the advantages of including duplicate traces. **The Examiner first argues that Rotenberg teaches duplicate traces may exist, and then argues that the combined references teach that the system of Rotenberg should not allow construction of duplicate traces. These arguments cannot coexist if the references are to teach the limitations of claim 5. If no duplicate traces can be constructed, there would be no reason to check the trace cache for a duplicate copy of a trace that the trace generator is constructing.**

2. The Examiner has not provided a valid reason to combine the references.

The Examiner submits that one of ordinary skill in the art at the time the invention was made would have been motivated to combine the teachings of Lange's checking for duplicates and discarding of the duplicates to Rotenberg in order to solve either or both of the problems of getting a miss while servicing a miss (in a system that allows duplicate traces), and preventing eviction of a useful trace by a duplicate trace (in a system that does not allow duplicate traces). These are clearly contradictory goals with contradictory solutions. In addition, as discussed above, Lange does not describe checking for a duplicate copy of a trace, as it has nothing to do with the operation of a trace cache.

The Examiner's reasons for combining the references are clearly not supported by the cited art, and the references, when combined, do not teach the limitation recited in claim 5.

For at least the reasons above, the rejection of claim 5 is unsupported by the cited art, and removal thereof is respectfully requested.

Claim 18 includes limitations similar to claim 5, and so the arguments presented above apply with equal force to this claim, as well.

Claims 6 and 19:

1. The cited art clearly fails to teach or suggest *wherein in response to the trace generator identifying a duplicate copy of the trace, the trace generator is configured to discard the trace under construction*, as recited in claim 6.

The Examiner cites Lange as teaching this limitation in column 5, lines 5-10. However, as discussed herein, Lange is not directed to trace generation at all. Lange's description of the operation of a data cache during fetching from main memory has

absolutely nothing to do with the specific limitations of Appellants' claims. For example, there is nothing in the combination of cited references that teaches or suggests identifying a duplicate copy of a trace and in response to such identification, discarding a trace under construction.

2. The Examiner has not provided any motivation or other reason to combine the references in teaching the specific limitations of claim 6. Therefore, the Examiner has failed to establish a *prima facie* obviousness of the claimed invention.

For at least the reasons above, the rejection of claim 6 is unsupported by the cited art, and removal thereof is respectfully requested.

Claim 19 includes limitations similar to claim 6, and so the arguments presented above apply with equal force to this claim, as well.

Claims 7 and 20:

1. The cited art clearly fails to teach or suggest *wherein in response to the trace generator identifying an entry corresponding to a duplicate copy of the trace, the trace generator is configured to check the trace cache for an entry corresponding to a next trace to be generated*, as recited in claim 7.

The Examiner asserts that Rotenberg teaches this limitation in Section 2.2, “The trace cache is checked every time there is a potential new trace, so when one trace is found and discarded, the next potential new trace will cause the trace cache to be checked again.” **Appellants assert that this clearly does not teach the specific limitations of Appellants' claim, as Rotenberg does not perform the checking in response to identifying a duplicate copy of the trace, as required by claim 7.** Instead, “the trace cache is checked every time there is a potential new trace” regardless of whether a duplicate copy has been identified.

2. The Examiner has not provided any motivation or other reason to combine the references in teaching the specific limitations of claim 7. Therefore, the Examiner has failed to establish a *prima facie* obviousness of the claimed invention.

For at least the reasons above, the rejection of claim 7 is unsupported by the cited art, and removal thereof is respectfully requested.

Claim 20 includes limitations similar to claim 7, and so the arguments presented above apply with equal force to this claim, as well.

Claims 8 and 21:

1. The cited art clearly fails to teach or suggest *wherein in response to the trace generator identifying a trace entry corresponding to the next trace to be generated, the trace generator is configured to discard the trace under construction*, as recited in claim 8.

The Examiner submits that Lange teaches this limitation in column 5, lines 5-10. However, as discussed above, Lange teaches nothing about the generation of trace entries, much less the specific limitations of claim 8. For example, there is nothing in the combination of references that teaches or suggests discarding a trace under construction in response to identifying a trace entry corresponding to the next trace to be generated (i.e., a trace other than the current trace entry).

2. The Examiner has not provided any motivation or other reason to combine the references in teaching the specific limitations of claim 8. Therefore, the Examiner has failed to establish a *prima facie* obviousness of the claimed invention.

For at least the reasons above, the rejection of claim 8 is unsupported by the cited art, and removal thereof is respectfully requested.

Claim 21 includes limitations similar to claim 8, and so the arguments presented above apply with equal force to this claim, as well.

Fifth ground of rejection:

The Examiner rejected claims 9 and 22 under 35 U.S.C. § 103(a) as being unpatentable over Rotenberg, Xia and Braught in view of Akkary, et al. (U.S. Patent 6,247,121) (hereinafter “Akkary”). Appellants traverse this rejection for at least the following reasons.

1. The cited art clearly fails to teach or suggest *wherein the trace generator is configured to generate traces in response to instructions being retired, as recited in claim 9.*

The Examiner admits that Rotenberg does not teach that instructions need to be retired before the trace can be generated and relies on Akkary to teach this limitation. The Examiner submits that Akkary teaches that instructions are not put into the trace buffers until they have been retired, to ensure that they executed correctly (column 3, lines 40-44). **This passage actually says just the opposite**, that is, that instructions placed in the trace buffer stay in the trace buffer until they are retired.

For at least the reasons above, the rejection of claim 9 is unsupported by the cited art, and removal thereof is respectfully requested.

Claim 22 includes limitations similar to claim 9, and so the arguments presented above apply with equal force to this claim, as well.

Sixth ground of rejection:

The Examiner rejected claims 27-31 and 35 under 35 U.S.C. § 103(a) as being unpatentable over Rotenberg, Xia, Braught and Lange in view of Akkary. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Independent Claims 27 and 35:

1. The cited art clearly fails to teach or suggest *receiving a retired instruction*.

The Examiner admits that Rotenberg does not teach that instructions need to be retired before the trace can be generated and relies on Akkary to teach this limitation. The Examiner submits that Akkary teaches that instructions are not put into the trace buffers until they have been retired, to ensure that they executed correctly (column 3, lines 40-44). **This passage actually says just the opposite**, that is, that instructions placed in the trace buffer stay in the trace buffer until they are retired.

2. The cited references fail to teach or suggest *in response to determining that a previous trace under construction duplicates a trace in a trace cache and that the received instruction corresponds to a branch label, beginning construction of a new trace*.

The Examiner admitted that Rotenberg fails to teach this limitation or to discuss the issue of duplication. The Examiner submits that Rotenberg discusses the disadvantages which occur when a trace cache miss occurs while servicing a previous trace cache miss, and teaches the disadvantages of a useless trace displacing a useful trace. The Examiner then submits, “Therefore, Rotenberg teaches a system in which allows tracing of potential duplicate traces, and also a system which requires action when a miss occurs while servicing a miss.” Appellants assert that the Examiner is

contradicting himself and that he has cited nothing in Rotenberg that teaches tracing of potential duplicate traces. In Rotenberg, the fill-line buffer logic services trace cache misses (i.e., the combination of a matching target address and matching branch flags is not found in the trace cache). There is no reason to believe there would ever be a duplicate trace in the system of Rotenberg, and no reasons or opportunity to avoid such duplication. The Examiner suggests that there is a “potential for duplicate traces to exist with path associativity in Rotenberg’s alternative embodiments.” **This is completely unsupported in Rotenberg and does nothing to overcome the deficiencies of the cited references in teaching the above-referenced limitations. Similarly, the Examiner’s contention that “Rotenberg indicates in his judicial trace selection that storing a duplicate trace would be at best useless, and at worst displace a useful trace that may be used” is also completely unsupported.** The solution discussed in this section is completely different from the specific limitations recited in claim 27. The Examiner submits that this section provides motivation to handle cases involving duplicate traces, and that Lange provides such a method, by simply discarding the duplicate value. **Again, the Examiner’s remarks do not address the limitations recited in claim 27, which are directed to specific conditions to be met before beginning construction of a new trace, and which the cited references do not teach.**

3. The Examiner has failed to provide a proper reason to combine the references.

The Examiner argues that Akkary suggests the limitation of using a retired instruction, and he includes remarks regarding “correctness” of a trace, a goal of Rotenberg’s trace cache to exploit code reuse, and that branches tend to be biased in one direction. However, Rotenberg explicitly states that the trace line is filled as instructions are fetched from the instruction cache, **which is clearly incompatible with filling the trace line with retired instructions.** The fact that the system of Rotenberg could solve this problem in a manner different from the only example described does not suggest the specific solution recited in the claims. **Appellants also assert that combining the references does not teach the claimed invention, as neither reference teaches a solution**

that involves receiving a retired instruction, and in response to determining that a previous trace under construction duplicates a trace in a trace cache and that the received instruction corresponds to a branch label, **beginning construction of a new trace.**

The Examiner previously cited Lange's description of the operation of a data cache during fetching from main memory. Appellants assert that this has absolutely nothing to do with the specific limitations of claim 27. Checking Rotenberg's trace cache for a duplicate trace before collecting a new trace is also contradictory to the Examiner's remarks above regarding the advantages of including duplicate traces. **The Examiner first argues that Rotenberg teaches duplicate traces may exist, and then argues that the combined references teach that the system of Rotenberg should not allow construction of duplicate traces. These arguments cannot coexist if the references are to teach the limitations of claim 27. If no duplicate traces can be constructed, there would be no reason to determine if a trace previously under construction was duplicating a trace already in the trace cache.** The Examiner's reasons for combining the references are clearly not supported by the cited art, and the references, when combined, do not teach the limitations recited in claim 27.

For at least the reasons above, the rejection of claim 27 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 35 includes limitations similar to claim 27, and so the arguments presented above apply with equal force to this claim, as well.

Claim 28:

1. The cited art clearly fails to teach or suggest *continuing construction of an incomplete trace already in process in response to determining that the incomplete trace does not duplicate a trace in a trace cache.*

The Examiner asserts that Rotenberg teaches this limitation in Section 2.2. However, as discussed above, the combination of cited references fails to teach or suggest checking the trace cache for a duplicate trace, much less the specific limitations recited in claim 28. In addition, the Examiner has not provided any arguments to support his assertion that this limitation is taught by Rotenberg.

2. The Examiner has not provided any motivation or other reason to combine the references in teaching the specific limitations of claim 28. Therefore, the Examiner has failed to establish a *prima facie* obviousness of the claimed invention.

For at least the reasons above, the rejection of claim 28 is unsupported by the cited art, and removal thereof is respectfully requested.

Claim 29:

1. The cited art clearly fails to teach or suggest *searching the trace cache for duplicate entries subsequent to completion of the previous trace under construction or the new trace*.

The Examiner submits that Lange teaches this limitation in column 5, lines 5-10. However, as discussed above, Lange teaches nothing about the operation of a trace cache, much less the specific limitations of claim 29. For example, there is nothing in the combination of cited references that teaches or suggests searching a trace cache for duplicate entries subsequent to completion of the previous trace under construction (i.e., that of claim 27) or the new trace. In addition, the Examiner has not provided any arguments to support his assertion that this limitation is taught by Lange.

2. The Examiner has not provided any motivation or other reason to combine the references in teaching the specific limitations of claim 29. Therefore,

the Examiner has failed to establish a *prima facie* obviousness of the claimed invention.

For at least the reasons above, the rejection of claim 29 is unsupported by the cited art, and removal thereof is respectfully requested.

Claim 30:

1. The cited art clearly fails to teach or suggest *creating a new entry in the trace cache in response to no duplicate entry being identified.*

The Examiner asserts that Rotenberg teaches this limitation in Section 2.2. However, as discussed above, the combination of cited references fails to teach or suggest checking the trace cache for a duplicate trace, much less the specific limitations recited in claim 28. In addition, the Examiner has not provided any arguments to support his assertion that this limitation is taught by Rotenberg.

2. The Examiner has not provided any motivation or other reason to combine the references in teaching the specific limitations of claim 30. Therefore, the Examiner has failed to establish a *prima facie* obviousness of the claimed invention.

For at least the reasons above, the rejection of claim 30 is unsupported by the cited art, and removal thereof is respectfully requested.

Claim 31:

1. The cited art clearly fails to teach or suggest *discarding a trace in response to a duplicate entry being identified.*

The Examiner submits that Lange teaches this limitation in column 5, lines 5-10. **However, as discussed above, Lange teaches nothing about the operation of a trace cache, much less the specific limitations of claim 31.** For example, there is nothing in the combination of cited references that teaches or suggests discarding a trace in response to a duplicate (trace cache) entry being identified. In addition, the Examiner has not provided any arguments to support his assertion that this limitation is taught by Lange.

2. The Examiner has not provided any motivation or other reason to combine the references in teaching the specific limitations of claim 31. Therefore, the Examiner has failed to establish a *prima facie* obviousness of the claimed invention.

For at least the reasons above, the rejection of claim 31 is unsupported by the cited art, and removal thereof is respectfully requested.

Seventh ground of rejection:

The Examiner rejected claims 32 - 34 under 35 U.S.C. § 103(a) as being unpatentable over Rotenberg in view of Xia. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 32, 33, and 34:

1. The cited art clearly fails to disclose *a method, comprising: fetching instructions from an instruction cache; continuing to fetch instructions from the instruction cache without searching a trace cache until a branch target address is generated; in response to a branch target address being generated, searching a trace cache for an entry corresponding to the branch target address.*

The Examiner submits that Rotenberg teaches all of the limitations of this claim except, “without searching a trace cache” and relies on Xia to teach this limitation. However, Rotenberg does not teach searching a trace cache in response to a branch target address being generated. **In fact, the Examiner admits that Rotenberg teaches that the trace cache is searched on every instruction**. Appellants note Xia’s trace table and instruction cache are also checked in parallel on every instruction. **The Examiner’s contention that it would be obvious not to search the trace cache for a non-branch instruction is completely unsupported, as his own references check the trace cache on every instruction**. His assertion that such a technique would contribute to power saving or critical path reduction is similarly unsupported in the cited art. The only advantage noted for starting traces on branches (i.e., to save memory space) does not require a change to the method for searching the trace cache. There is nothing in Xia that describes how, or more importantly when, the trace cache is checked for the start of a trace line, or when the system begins a search of the trace cache for any reason. The Examiner’s statement, “The advantages that Examiner indicated... are obvious advantages one of ordinary skill in the art would recognize, searching a cache takes time and energy, and if there is no possible way that searching a cache can benefit the user, there is absolutely no reason to do so” contradict the teachings of his own references, which search the cache for a hit (although not necessarily a trace start) on every cycle. **This is in direct conflict with the limitations of claim 32.**

For at least the reasons above, the rejection of claim 32 is unsupported by the cited art, and removal thereof is respectfully requested.

CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-33 was erroneous, and reversal of his decision is respectfully requested.

Since this Appeal Brief is submitted within one month from the Notice of Panel Decision, no extension of time fee should be due. The Commissioner is authorized to charge the appeal brief fee and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5500-88700/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: January 7, 2008

VIII. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A microprocessor, comprising:

an instruction cache configured to store instructions;

a branch prediction unit;

a trace cache configured to store a plurality of traces of instructions; and

a prefetch unit coupled to the instruction cache, the branch prediction unit, and the trace cache;

wherein the prefetch unit is configured to fetch instructions from the instruction cache until the branch prediction unit outputs a predicted target address;

wherein the prefetch unit is configured to check the trace cache for a match for the predicted target address in response to the branch prediction unit outputting the predicted target address;

wherein the prefetch unit is configured to not check the trace cache for a match until the branch prediction unit outputs the predicted target address; and

wherein in response to the prefetch unit identifying a match for the predicted target address in the trace cache, the prefetch unit is configured to fetch one or more of the plurality of traces from the trace cache.

2. The microprocessor of claim 1, wherein the branch prediction unit is configured to output the predicted target address in response to a prediction that a branch will be taken.

3. The microprocessor of claim 1, wherein the branch prediction unit is configured to output the predicted target address in response to detection of a branch misprediction.

4. The microprocessor of claim 1, further comprising a trace generator, wherein the trace generator is configured to begin a trace with an instruction corresponding to a label boundary.

5. The microprocessor of claim 4, wherein the trace generator is configured to check the trace cache for a duplicate copy of the trace that the trace generator is constructing.

6. The microprocessor of claim 5, wherein in response to the trace generator identifying a duplicate copy of the trace, the trace generator is configured to discard the trace under construction.

7. The microprocessor of claim 5, wherein in response to the trace generator identifying an entry corresponding to a duplicate copy of the trace, the trace generator is configured to check the trace cache for an entry corresponding to a next trace to be generated.

8. The microprocessor of claim 7, wherein in response to the trace generator identifying a trace entry corresponding to the next trace to be generated, the trace generator is configured to discard the trace under construction.

9. The microprocessor of claim 4, wherein the trace generator is configured to generate traces in response to instructions being retired.

10. The microprocessor of claim 4, wherein the trace generator is configured to generate traces in response to instructions being decoded.

11. The microprocessor of claim 1, wherein each of the plurality of traces comprises partially-decoded instructions.

12. The microprocessor of claim 1, wherein each of the plurality of traces is associated with a tag comprising the address of an earliest instruction, in program order, stored within that trace.

13. The microprocessor of claim 1, wherein each of the plurality of traces is associated with a flow control field comprising a label for an instruction to which control will pass for each branch operation comprised in that trace.

14. A computer system, comprising:

a system memory; and

a microprocessor coupled to the system memory, comprising:

an instruction cache configured to store instructions;

a branch prediction unit;

a trace cache configured to store a plurality of traces of instructions; and

a prefetch unit coupled to the instruction cache, the branch prediction unit,
and the trace cache;

wherein the prefetch unit is configured to fetch instructions from the instruction cache until the branch prediction unit outputs a predicted target address;

wherein the prefetch unit is configured to check the trace cache for a match for the predicted target address in response to the branch prediction unit outputting the predicted target address;

wherein the prefetch unit is configured to not check the trace cache for a match until the branch prediction unit outputs the predicted target address; and

wherein in response to the prefetch unit identifying a match for the predicted target address in the trace cache, the prefetch unit is configured to fetch one or more of the plurality of traces from the trace cache.

15. The computer system of claim 14, wherein the branch prediction unit is configured to output the predicted target address in response to a prediction that a branch will be taken.

16. The computer system of claim 14, wherein the branch prediction unit is configured to output the predicted target address in response to detection of a branch misprediction.

17. The computer system of claim 14, further comprising a trace generator, wherein the trace generator is configured to begin a trace with an instruction corresponding to a label boundary.

18. The computer system of claim 17, wherein the trace generator is configured to check the trace cache for a duplicate copy of the trace that the trace generator is constructing.

19. The computer system of claim 18, wherein in response to the trace generator identifying a duplicate copy of the trace, the trace generator is configured to discard the trace under construction.

20. The computer system of claim 18, wherein in response to the trace generator identifying an entry corresponding to a duplicate copy of the trace, the trace generator is configured to check the trace cache for an entry corresponding to a next trace to be generated.

21. The computer system of claim 20, wherein in response to the trace generator identifying a trace entry corresponding to the next trace to be generated, the trace generator is configured to discard the trace under construction.

22. The computer system of claim 17, wherein the trace generator is configured to generate traces in response to instructions being retired.

23. The computer system of claim 17, wherein the trace generator is configured to generate traces in response to instructions being decoded.

24. The computer system of claim 14, wherein each of the plurality of traces comprises partially-decoded instructions.

25. The computer system of claim 14, wherein each of the plurality of traces is associated with a tag comprising the address of an earliest instruction, in program order, stored within that trace.

26. The computer system of claim 14, wherein each of the plurality of traces is associated with a flow control field comprising a label for an instruction to which control will pass for each branch operation comprised in that trace.

27. A method, comprising:

receiving a retired instruction;

determining if a previous trace under construction duplicates a trace in a trace cache and if the received instruction corresponds to a branch label; and

in response to determining that a previous trace under construction duplicates a trace in a trace cache and that the received instruction corresponds to a branch label, beginning construction of a new trace.

28. The method of claim 27, further comprising continuing construction of an incomplete trace already in process in response to determining that the incomplete trace does not duplicate a trace in a trace cache.

29. The method of claim 27, further comprising searching the trace cache for duplicate entries subsequent to completion of the previous trace under construction or the new trace.

30. The method of claim 29, further comprising creating a new entry in the trace cache in response to no duplicate entry being identified.

31. The method of claim 29, further comprising discarding a trace in response to a duplicate entry being identified.

32. A method, comprising:

fetching instructions from an instruction cache;

continuing to fetch instructions from the instruction cache without searching a trace cache until a branch target address is generated;

in response to a branch target address being generated, searching a trace cache for an entry corresponding to the branch target address.

33. The method of claim 32, further comprising continuing to fetch instructions from the instruction cache in response to no entry being identified in the trace cache corresponding to the branch target address.

34. The method of claim 32, further comprising fetching one or more traces from the trace cache in response to an entry being identified in the trace cache corresponding to the branch target address.

35. A microprocessor, comprising:

means for receiving a retired operation;

means for determining if a previous trace under construction duplicates a trace in a trace cache and if the received operation is a first operation at a branch label; and

means for starting a new trace in response to determining that a previous trace under construction duplicates a trace in a trace cache and that the received operation is a first operation at a branch label.

IX. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.